Time-Series Data Exchange Using the Geography Markup Language

Daniel R. Martin[1], John Ulmer[2] and James Boyd[3]
[1] I.M. Systems Group, Charleston, SC 29405
[2] Perot Systems, Charleston, SC 29405
[3] NOAA Coastal Services Center, Charleston, SC 29405

***Abstract* – The Geography Markup Language (GML) is a well known XML dialect that offers the ocean observing community a robust and standards based encoding for data exchange. Adaptations of GML are evolving to meet the needs and uses of different communities. Understanding how to use GML to model and exchange observatory time-series data will be critical to building a national data resource from the federation of observatories that form the Integrated Ocean Observing System (IOOS). The NOAA Coastal Services Center conducted an experiment with several nonfederal partners using the GML Simple Features Profile and the Web Feature Service (WFS). An application schema, a content model, a GML dictionary, and a special purpose WFS were used to exchange data between an observatory and a data-assembly center relational database.**

I. Introduction

Nonfederal ocean observatories make up a broad community of data provides that need to exchange their data within the Integrated Ocean Observing System (IOOS). Nonfederal assets increase the resolution and frequency of data and information products, thus helping to apply IOOS both locally and nationally. Much of the nonfederal data is characterized as time-series observations from in situ or mobile platforms. These data are exchanged among the federation of providers and consumers using a spectrum of technologies. For several years the Ocean.US has sponsored expert teams and provided guidance to concentrate and foster data exchange through a standards-based approach. The Open Geospatial Consortium (OGC) Geography Markup Language (GML) [1] and Web Feature Service (WFS) [2] are part of this family of relevant standards. Efforts to understand and implement GML and WFS between nonfederal and federal communities are relatively new and few. The NOAA Coastal Services Center in cooperation with the IOOS Program Office has designed and begun conducting experimental tests using GML and WFS with several nonfederal observatories.

GML contains a rich library of components used to model and encode data for exchange within a service-based architecture. It is harmonized to many of the ISO 19100 series standards and can be specialized for specific disciplines. For the loose federation of providers that make up IOOS, GML and application schemas offer the potential for large increases in interoperability.

We explain in this paper one experimental implementation of GML and the WFS transport mechanism. Our scope is based on simple scalar time-series records and the need to meet the requirements of data exchange between an observatory and a data-assembly center that specializes in integrating and hosting geospatial time-series content. We cover the building of a source and destination database, a record content standard, an application schema, our service application and performance measurements.

II. Observatory and Data Assembly Center Storage

Two storage configurations are used in our experiment to characterize each end of the transport environment. Both use a relational database management system. The observation database is the first persistent storage facility for sensor records after data collection. The observation geodatabase is a dedicated storage facility, housed in a data-assembly center where it receives input from geographically diverse observation databases. By understanding the design of each database we can determine the content and structures necessary in the transport tier.

The observation database operates as a data warehouse storing all records associated with data collection, quality, operations, archiving and distribution at a geographically constrained scale. A common design for these records does not yet exist. We developed a new database design which was implemented in Microsoft SQL Server and PostgreSQL as shown in the entity relationship diagram in Fig. 1. The goal of the design was to support

0-933957-35-1

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **SEP 2007**	2. REPORT TYPE	3. DATES COVERED **00-00-2007 to 00-00-2007**

4. TITLE AND SUBTITLE **Time-Series Data Exchange Using the Geography Markup Language**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **I.M. Systems Group,Charleston,SC,29405**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM002047. Presented at the MTS/IEEE Oceans 2007 Conference held in Vancouver, Canada on Sep 29-Oct 4, 2007. U.S. Government or Federal Purpose Rights License.

14. ABSTRACT
See Report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**7**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

- Current and historic records of platform and observed values
- In situ observations with extension to mobile and other vector values
- Deployment on a variety of RDBMS configurations with minimal modifications
- Support for geospatial queries

Content for the database was harvested from the NOAA National Data Buoy Center file system. Operational data from more than five years were extracted and loaded into the database for stations in the Atlantic and Gulf of Mexico. Approximately 100 unique locations were recorded each collecting between 5 and 10 observed parameters. The frequency of observations is variable but usually no more than at a 10 minute rate. Observed parameters included water temperature, currents, water level, waves and meteorological values. We found that time based queries from tables with millions of records worked very well, and the use of views was an excellent mechanism to loosely couple the data storage from the transport tier.

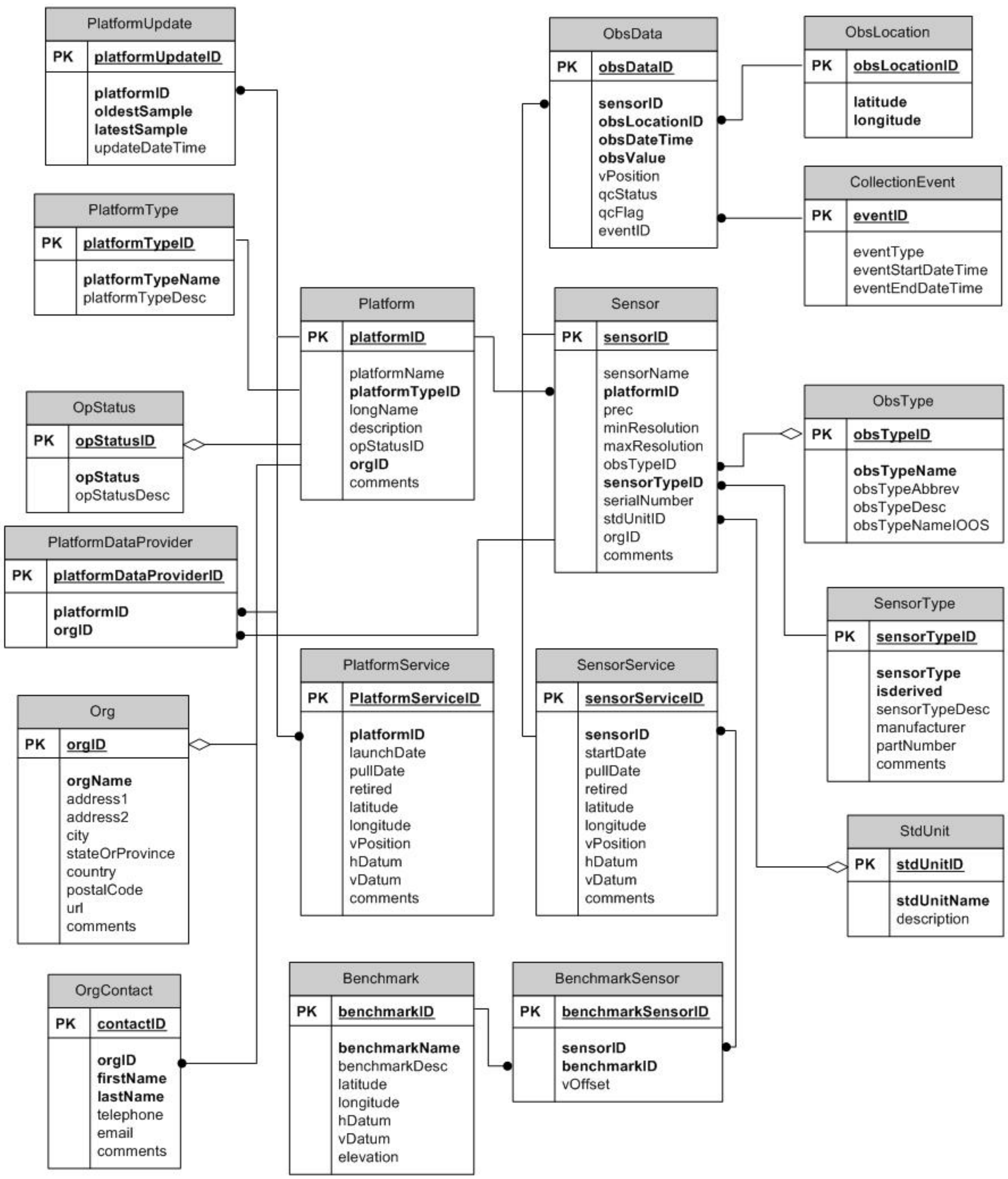

Fig. 1 Observation Database Entity-Relationship Diagram

The observation geodatabase operates in a data-assembly center. It provides a subset of features specially managed for aspects of time and geospatial content and for ease of access by end users. The observation geodatabase is loosely based on sections of the ArcHydro [3] and Marine Data Model [4] shown in Fig 3.

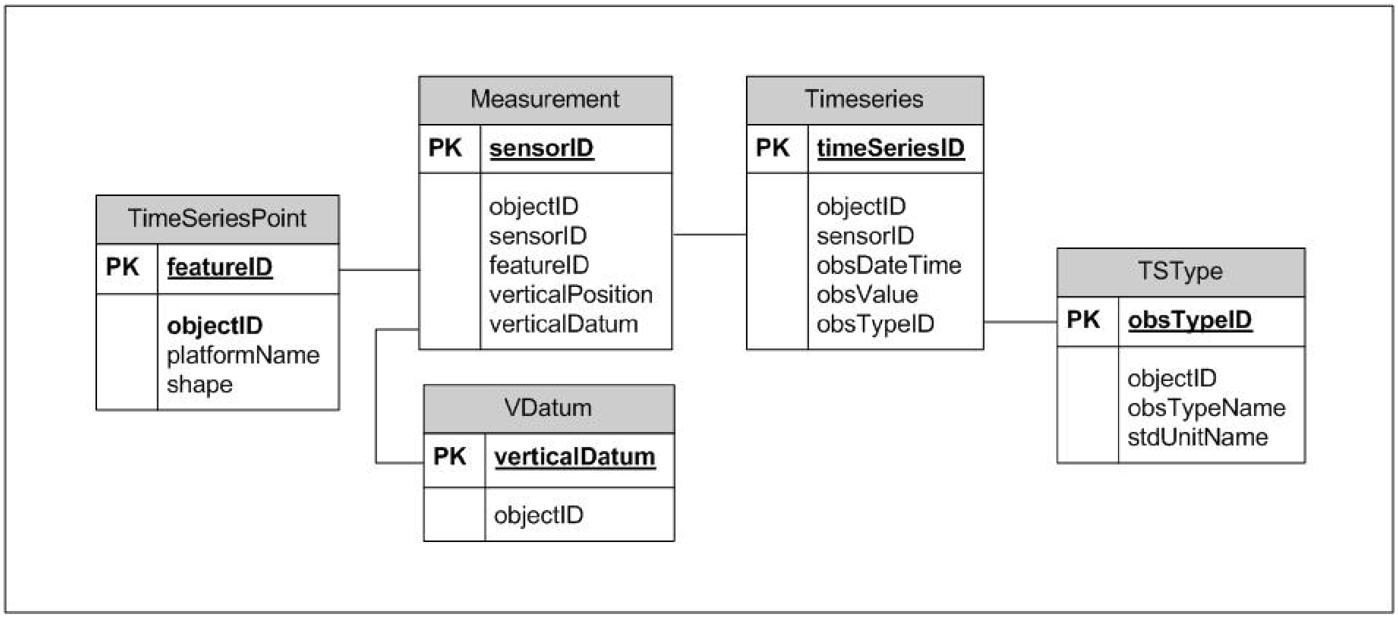

Fig. 2 Observation Geodatabase Entity-Relationship Diagram

One implementation was deployed using SQL Server Express and ArcGIS. A benefit of this design is that a relationship can be maintained between a single horizontal location and a collection of time-series values at *n* number of vertical positions at that location. This approach eliminates the duplication of horizontal and vertical measurements. Using a spatial database extension also allows for querying time-based values in conjunction with spatial filters. Through the process of building and deploying the observation and geodatabase we were able to identify a general class of content and data-structure requirements for transport.

III. DATA CONTENT

Effective sharing and utilization of data requires that data providers and data consumers have a common understanding of the meaning and structure of each data package exchanged. In the geospatial community organizations such as the Federal Geographic Data Committee have supported activities to define content standards and application schemas for seven major data themes of national significance. The content standards define the precise scope and definition of the elements in a logical record. The application schemas contain the rules that define the record structure, including cardinality, types, relationships, and domains, and they provide a mechanism for validation. Well-defined and accepted content standards and application schemas do not currently exist among observatories and their consumers. As a result data consumers often need to recast data to normalize units of time, location, units of measure and the physical format for their own use.

We defined an original content model for the purpose of data exchange between the observatory and the data-assembly center. Our goal was to include the minimum elements required to define location, time, parameter and original source. We also wanted a simple model that would ease aggregation of records and meet the needs of a broad spectrum of users. Elements from the IOOS Observation Registry [5] were reviewed as source material. Although most respondents' record content did not match exactly, the similarities were considerable. For a parameter such as water temperature for which a simple single scalar value is needed, we were able to define a common record with the content listed in table I.

We recognized that in contrast to the rich set of elements in the observation database this content model is minimal. However this approach can work when metadata exist and a key to original source can be maintained. We maintained a linkage with the observation database by building a composite key using the observation database identity fields of organizationID and

platformID. In this context our definition of a platform is identical to collection device, sensor or instrument. We also included a platform name field that uniquely identifies each platform as designated by the collecting observatory. For more complex observations such as currents, which require both vector and magnitude to be reported, this model can be extended with additional sequences of name, unit and value. This same content can be generically applied to observations derived from mobile platforms.

TABLE I
RECORD CONTENT

Parameter	Value	Definition
Platform ID	312	Database identity
Organization ID	12	Database identity
Platform Name	A001.SensorABC	Observatory designation
Latitude	42.123	Decimal Degrees, positive values for North
Longitude	-72.234	Decimal Degrees, negative values for West
Horizontal Reference	WGS 84	WGS 84
Vertical Position	6.4	Decimal meter, positive values above datum
Vertical Reference	MHW	Reference GML dictionary
Date – Time	2007-10-04T12:01:02Z	ISO 8601
Observation Name	waterTemperature	Reference GML dictionary
Observation Unit	Celsius	Reference GML dictionary
Observation Value	18.34	

The definition of each element in the content model was fixed for the scope of the test. We reviewed several sources of semantic content and interoperability mechanisms: NASA Semantic Web for Earth and Environmental Terminology, NetCDF Climate and Forecast Metadata Convention, Marine Metadata Interoperability and the National Oceanographic Data Center. We decided to build GML simple dictionaries which proved to be easily deployed containers that could host content from any of the semantic sources that we found; they were also our best option to avoid using explicit remote references. We were also able to select just the content relevant to our domain, thus significantly reducing the complexity of the vocabulary. Definitions for time and location were defined as shown in table I since we could not find a standard at either the agency or community level. We adopted a variation of the OGC Universal Resource Name (URN) convention in our dictionaries to designate the namespace for the vocabularies and units. Although the use of the URN convention was not necessary to complete our tests, the practice of using it provided a greater level of control and clarity to our vocabulary.

IV. DATA STRUCTURE

Within the family of OGC and similar specifications are different frameworks for encoding structure and content. Most approaches are based on either a constraint or extension to native GML, or have strong similarities to the GML and XML Schema. Some of these include Observation and Measurement, Climate Science Modeling Language, GeoRSS, Simple Features, Simple Features Profile and Keyhole Markup Language. We selected the GML Simple Features Profile level 1 specification [6] which lends itself well to delivery using the Web Feature Service, contains the necessary spatial and none-spatial property types, and is easy to learn and adapt. Using XML Schema [7], we defined a single application schema for the property, element and name definitions in our record as shown by Fig. 3. The basic structure is characterized by a single complex type named insituTimeSeries which contains six elements - five of them static “station” information, such as sensor and observation name. The sixth element is a time-series property type an unbounded sequence that contains date time and the observation value.

```
<xs:element name="TSMeasurement">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="obsDateTime" type="xs:dateTime"/>
   <xs:element name="observation" type="gml:MeasureType"/>
  </xs:sequence>
 </xs:complexType>
</xs:element>

<xs:element name="sensor">
 <xs:complexType>
   <xs:simpleContent>
```

```
      <xs:restriction base="gml:CodeType">
       <xs:attribute name="codeSpace" type="xs:anyURI" use="optional"
     default="http://csc.noaa.gov/ioos/dictionaries/SensorDictionary.xml"/>
      </xs:restriction>
     </xs:simpleContent>
   </xs:complexType>
  </xs:element>
  <xs:element name="observationName">
   <xs:complexType>
    <xs:simpleContent>
     <xs:restriction base="gml:CodeType">
      <xs:attribute name="codeSpace" type="xs:anyURI" use="optional"
        default="http://csc.noaa.gov/ioos/dictionaries/PhyOceanDictionary.xml"/>
     </xs:restriction>
    </xs:simpleContent>
   </xs:complexType>
  </xs:element>
  <xs:element name="verticalDatum">
   <xs:complexType>
    <xs:simpleContent>
     <xs:restriction base="gml:CodeType">
      <xs:attribute name="codeSpace" type="xs:anyURI" use="optional"
        default="http://csc.noaa.gov/ioos/dicitonary/VerticalDatumDicitonary.xml"/>
     </xs:restriction>
     </xs:simpleContent>
   </xs:complexType>
  </xs:element>
  <xs:element name="verticalPosition" type="gml:MeasureType"/>
  <xs:element name="horizontalPosition" type="gml:PointPropertyType"/>
  <xs:element name="tsEvent" type="ioos:TSMeasurementPropertyType" maxOccurs="unbounded"/>
```

Fig. 3 Application Schema Fragment

V. THE DATA SERVICE

Data exchange between nonfederal observatories and a federal data-assembly center has to be based on a protocol for loosely bound relationships and highly mixed operational practices. Two conceptual designs for IOOS specify a service- oriented architecture and web services as a strategy in this setting [8] and [9]. The OGC WFS is one specific implementation of a web service frequently used in the geospatial community to exchange base geospatial features, often static and defined in a single table structure. In an operational deployment, a data provider would host a WFS application on their web server. The application would receive data requests from a client, submit queries to a database and format and deliver GML encoded records back to the client. Time-series data encoding and its pattern of data exchange is different from those usually implemented for geospatial features. Our goal was to determine through this experiment if the WFS could support a different data model and use pattern typical of time-series data.

The current version of WFS, 1.1, is loosely tied to the GML 3.1.1 that includes the Simple Features Profile used for our encoding. At the time of our experiment, almost no commercial or Open Source implementations of WFS supported version 1.1 and the GML 3.1.1. We also found that many of them were burdened with additional libraries for map display, transactions, specialized data sources and other unnecessary components. Without substantial reworking, none would support our user-defined schema. We decided to develop an experimental WFS getFeature operation engineered for our data source and application schema. The getFeature operation is one of the three mandatory operations of standard WFS. We developed both a Perl and a Java release in approximately 400 lines of code each. Three filters were implemented to query the spatial extent, a time frame and the observation parameter. Although we were able to meet only the core specifications of the WFS getFeature operation during the experiment, we found that the interface standards of WFS easy supported the time-series request patterns and our output schema.

VI. CLIENT APPLICATIONS

The client in this data-exchange experiment is the data-assembly center. For our purpose, the only requirement for the client was to issue the WFS request to an observatory data service and to consume the GML-encoded response. We built a collection of WFS calls that were issued using a web page, a web-browser address bar and our performance-monitoring software. All requests used the HTTP GET protocol. An example request is shown in Fig. 5.

```
http://csc-s-ial-p.csc.noaa.gov/cgibin/microwfs/microWFS.cgi?
SERVICENAME=dtlservice&
REQUEST=getFeature&
SERVICE=microWFS&
VERSION=1.1.0&
OUTPUTFORMAT=text/xml;subType=gml/3.1.1/profiles/gmlsf/1.0.0/1&
BBOX=-69.00,32.0,-72.00,42.30&
TIME=2007-06-01T12:00Z,2007-06-01T14:00Z&
TYPENAME=waterTemperature
```

Fig. 4 WFS Request

A more advanced data assembly center client would schedule, manage and monitor requests and responses as well as parse and load records for storage and integration. The complexity of this parsing process is directly related to the record encoding defined by the schema and by the variability of the data. The experimental time-series record was relatively simple, and was easily ingested even using a general purpose desktop client such as MS Excel. A fragment of the response is shown in Fig. 5.

```
<gml:featureMember>
 <ioos:insituTimeSeries gml:id="ID000001">
  <ioos:sensor codeSpace="urn:x-noaa:source:insitu:NFRA:2007a">A001</ioos:sensor>
  <ioos:observationName
   codeSpace="urn:x-noaa:def:phyOcean:2007a">waterTemperature</ioos:observationName>
  <ioos:verticalDatum codeSpace="urn:x-noaa:def:verticalDatum">MLLW</ioos:verticalDatum>
  <ioos:verticalPosition uom="urn:x-noaa:def:uom:2007a:meter">1.4</ioos:verticalPosition>
  <ioos:horizontalPosition>
   <gml:Point>
    <gml:pos>60.59 -146.83</gml:pos>
   </gml:Point>
  </ioos:horizontalPosition>

 <ioos:tsEvent>
  <ioos:TSMeasurement>
   <ioos:obsDateTime>2001-12-17T09:30:47.0Z</ioos:obsDateTime>
   <ioos:observation uom="urn:x-noaa:def:uom:2007a:celsius">15.123</ioos:observation>
  </ioos:TSMeasurement>
 </ioos:tsEvent>
 <ioos:tsEvent>

 ...additional time series records continue here...

 </ioos:insituTimeSeries>
</gml:featureMember>
...additional featureMember records continue here...
</gml:FeatureCollection>
```

Fig. 5 Record Fragment

VII. Performance Metrics

Performance metrics determined application and database server-response times for common request patterns in our experiment. The Apache JMeter application [10] was used on the client and the SQL Server Profiler was used on the database host. The database host had 4, 3GHz CPUs, and 4GB RAM with local and network storage; the application host was similar but with 2 CPUs. Two general test cases were run.

In the first set, a query was submitted to return an hour's worth of data from 21 stations. Eighty-four time events were exchanged; each payload comprised approximately 36 Kilobytes. A simulation was executed with this query and a load of 1 user, 10 users, and 100 users. Response times ranged between 1 and 10 seconds; no noticeable load was detected on either host. Database response rates ranged between 35 and 75 milliseconds.

The second set of tests expanded the query to include 3 hours of observation data, resulting in a payload of 88 Kilobytes. Response times increased to a range of 3 to 12 seconds. The CPU usage rate for the database server increased to a range between 12 and 95 percent. Database response times ranged from 85 ms to 150 ms.

These metrics indicate acceptable performance for this configuration with a large margin for expansion in the number of parameters reported by an observatory. Data exchange strictly between an observatory and a data assembly center is likely to have a user load that would remain low and regular, with queries seeking only values very near real time. This use pattern indicates an even more favorable setting for a GML and WFS data-exchange deployment.

VIII. Conclusions and future work

For a nonfederal observatory examining their data publishing options, the experimental results indicate that the GML Simple Features Profile and WFS offer robust interoperability and a potential to lead towards new data products at the regional and national scale. For a data assembly center to build these new products from such a large federation of observatories, content standards and schemas must be developed and widely adopted. Some of this work is already underway through activities like the OGC Oceans Interoperability Experiment and the GALEON Project and showing some success.

The database designs, content standard, dictionaries, schemas and applications used in this experiment were very rapidly developed and tested among a small volunteer community of observatories and the Coastal Services Center. A closer look at these tools and standards with other parameters, platforms and users would help to determine their operational viability.

Acknowledgment

The authors wish to thank John Bercik of I.M. Systems Group and Yanlin Ye of Perot Systems. Funding support for this work has been provided by the National Oceanic and Atmospheric Administration Coastal Services Center, Award #EA133CO4CN0044.

References

[1] S. Cox, P. Daisey, R. Lake, C. Portele, A. Whiteside (eds.), “Geography Markup Language (GML) Implementation Specification, version 3.1.1”, Open Geospatial Consortium Inc., January 2005.
[2] P. Vretanos (ed.), “Web Feature Service Implementation Specification, version 1.1.0” Open Geospatial Consortium Inc., May 2005.
[3] D. Maidment (ed.), “Arc Hydro GIS for Water Resources”, ESRI Press, 2002.
[4] Wright et al. “Marine Data Model”, http://dusk.geo.orst.edu/djl/arcgis/index.html., 2001.
[5] IOOS Observation Registry Home Page, http://oceanobs.org/wc/map/., 2007
[6] P. Vretanos (ed.) “Geography Markup Language (GML) simple features profile Version 1.0”, Open Geospatial Consortium Inc., August 2006.
[7] W3C XML Schema Working Group, “XML Schema Definition Language: 1.0”, http://www.w3.org/XML/Schema.
[8] Raytheon Corporation, “Volume 1 – IOOS Conceptual Design”, August 2006.
[9] Lockheed Martin, “Integrated Ocean Observing System Conceptual Design”, September 2006.
[10] Apache JMeter, Apache Software Foundation, http://jakarta.apache.org/jmeter/.